

Admins - Demande #4994

[pad.april.org] Partition mysql presque pleine

04/12/2020 13:57 - Christian P. Momon

Statut:	Fermé	Début:	04/12/2020
Priorité:	Normale	Echéance:	
Assigné à:	François Poulain	% réalisé:	0%
Catégorie:		Temps estimé:	0.00 heure
Version cible:	Été 2021	Temps passé:	0.00 heure
Difficulté:	2 Facile		

Description

Sur Freenode#april-admin :

```
11:57 < PoluX> !list
11:57 < pascontent[14]> volume /var/lib/mysql (1): pad
12:04 < QGuLL> PoluX: /dev/mapper/vg_pad_data-mysql 7,8G 6,7G 829M 90% /var/lib/mysql
12:04 < QGuLL> pas glop
12:06 < PoluX> QGuLL: innodb qui grossi ?
12:07 < QGuLL> j'ai pas investigué encore
12:08 < PoluX> 4,5GI/var/lib/mysql/etherpad/store.ibd
12:08 < PoluX> or 225MI/var/backups/mysql/etherpad.sql.bz2
12:09 < QGuLL> c'est louche
12:10 < PoluX> beh ça me rappelle un bug connu
12:10 < PoluX> qui se résout par un dump / drop / restore
12:20 < PoluX> j'essaie un optimize table store;
12:37 < cpm_screen> PoluX: l'export SQL de la base etherpad fait 2,8 Go, donc la moitié de ce qui est occupé en base, voui, c'est louche
```

Historique

#1 - 04/12/2020 13:58 - Christian P. Momon

- Statut changé de Nouveau à En cours de traitement
- Assigné à mis à François Poulain
- Version cible changé de Backlog à Décembre 2020

Sur Freenode#april-admin :

```
12:38 < PoluX> -> ;
12:38 < PoluX> +-----+-----+-----+-----+
-----+
12:38 < PoluX> | Table          | Op          | Msg_type | Msg_text
|
12:38 < PoluX> +-----+-----+-----+-----+
-----+
12:38 < PoluX> | etherpad.store | optimize | note     | Table does not support optimize, doing recreate + anal
yze instead |
12:38 < PoluX> | etherpad.store | optimize | error    | The table 'store' is full
|
12:38 < PoluX> | etherpad.store | optimize | status   | Operation failed
|
12:38 < PoluX> +-----+-----+-----+-----+
-----+
12:38 < PoluX> 3 rows in set, 1 warning (15 min 43.310 sec)
12:38 < PoluX> cpm_screen: amha ça ira plus vite de couper le service et faire un dump/drop/restore
12:39 < PoluX> cf https://dba.stackexchange.com/questions/24942/how-do-i-shrink-the-innodb-file-ibdata1-withou
t-dumping-all-databases/24963#24963
12:39 < PoluX> c'est une vieille maladie de mysql
12:39 < PoluX> on a déjà fait ça à l'april
12:42 < cpm_screen> une occasion de valider la sauvegarde ;)
12:44 -!- pascontent[15] is now known as pascontent[14]
12:45 < PoluX> beh le dump sql c'est fiable, éprouvé
```

12:45 < PoluX> allez ; je coupe le pad et je fais la manip le temps du repas

#2 - 04/12/2020 14:01 - François Poulain

En résumé:

```
systemctl stop etherpad-lite.service
mysqldump -u pad -pbling etherpad | gzip > /var/tmp/20201204_etherpad.sql.gz
mysql -u pad -pbling etherpad -e 'drop database etherpad;'
mysql -u pad -pbling -e 'create database etherpad;'
zcat /var/tmp/20201204_etherpad.sql.gz | mysql -u pad -pyuCi2aco etherpad
systemctl start etherpad-lite.service
```

#3 - 04/12/2020 15:53 - François Poulain

Au final la table store est quand même énorme.

Je me suis amusé à la chose suivante:

1. Trouver des timestamps dans le dump.

```
zgrep -bo '"timestamp\":[0-9]\+' /var/tmp/20201204_etherpad.sql.gz | gzip > 20201204_etherpad_timestamps.gz
```

2. Un petit coup de sed

```
gzip -d 20201204_etherpad_timestamps.gz
sed -i -e 's/"timestamp\":[0-9]\+/' 20201204_etherpad_timestamps
```

3. Impossible de rentrer ça dans un logiciel simple de stats genre libreoffice tellement y en a

4. Simplement on va regarder les occurrences période par période :

```
$ for ts in $(seq 130 165); do date -d @"$ts"0000000 && grep -c ",$ts" 20201204_etherpad_timestamps ; done
dim. 13 mars 2011 08:06:40 CET
0
jeu. 07 juil. 2011 02:53:20 CEST
0
dim. 30 oct. 2011 19:40:00 CET
0
jeu. 23 févr. 2012 13:26:40 CET
0
lun. 18 juin 2012 08:13:20 CEST
365
ven. 12 oct. 2012 02:00:00 CEST
132176
lun. 04 févr. 2013 18:46:40 CET
112352
ven. 31 mai 2013 13:33:20 CEST
276829
mar. 24 sept. 2013 07:20:00 CEST
243830
sam. 18 janv. 2014 00:06:40 CET
149711
mar. 13 mai 2014 18:53:20 CEST
118833
sam. 06 sept. 2014 12:40:00 CEST
131974
mer. 31 déc. 2014 05:26:40 CET
84866
dim. 26 avril 2015 00:13:20 CEST
103349
mer. 19 août 2015 18:00:00 CEST
151989
dim. 13 déc. 2015 10:46:40 CET
151046
jeu. 07 avril 2016 05:33:20 CEST
121106
dim. 31 juil. 2016 23:20:00 CEST
108158
jeu. 24 nov. 2016 16:06:40 CET
129060
lun. 20 mars 2017 09:53:20 CET
164102
```

ven. 14 juil. 2017 04:40:00 CEST
139785
lun. 06 nov. 2017 21:26:40 CET
123623
ven. 02 mars 2018 15:13:20 CET
188773
mar. 26 juin 2018 10:00:00 CEST
148420
sam. 20 oct. 2018 03:46:40 CEST
161310
mar. 12 févr. 2019 20:33:20 CET
165314
sam. 08 juin 2019 15:20:00 CEST
103167
mer. 02 oct. 2019 09:06:40 CEST
141548
dim. 26 janv. 2020 01:53:20 CET
3687960
mer. 20 mai 2020 20:40:00 CEST
821167
dim. 13 sept. 2020 14:26:40 CEST
356578
jeu. 07 janv. 2021 07:13:20 CET
0
lun. 03 mai 2021 02:00:00 CEST
0
jeu. 26 août 2021 19:46:40 CEST
0
lun. 20 déc. 2021 12:33:20 CET
0
ven. 15 avril 2022 07:20:00 CEST
0

Moralité : un truc chelou est arrivé en 2020.

#5 - 04/12/2020 17:02 - François Poulain

Proposition:

1. Coller dans les gros pads qui sont pas April le message suivant :

«
Bonjour,

Le service pad.april.org n'est pas destiné à accueillir du contenu qui ne concerne pas l'April.
Nous avons des soucis de performance sur ce service et donc souhaitons le délester.

Si vous avez besoin d'un pad public, vous pouvez utiliser celui de <https://pad.chapril.org>

Cordialement.
François Poulain, pour les administrateurs système de l'April
contact@april.org
»

Puis dégager les pads au bout de qq semaines

2. Conserver le dump de la db puis dans qq semaines, virer de la db de prod tous les sessionStorage qui sont dans la db conservée.

#6 - 05/12/2020 12:11 - Christian P. Momon

François Poulain a écrit :

Moralité : un truc chelou est arrivé en 2020.

Le confinement ;o)

Blague à part, j'ai repris tes commandes :

```
grep -o '\\\"timestamp\\\":[0-9]\\{10\\}' etherpad.sql | awk -F: '{print $2}' > timestamps
sort -n timestamps > sortedtimestamps
# La commande suivante prenant plusieurs heures, en fait, j'ai splité pour paralléliser...
while read line; do date -d @"$line" +%Y-%m; done < sortedtimestamps > yearmonth
uniq -c yearmonth
```

```
cpm@ocmstar (11:57:09) ~/Téléchargement/XX 126 > uniq -c yearmonth
 6739 2019-03
 50490 2019-04
 37899 2019-05
 35076 2019-06
 19067 2019-07
 25790 2019-08
 34820 2019-09
 28131 2019-10
 33646 2019-11
 50115 2019-12
 40587 2020-01
 53133 2020-02
1472622 2020-03
1763193 2020-04
 495887 2020-05
 426726 2020-06
155931 2020-07
 75812 2020-08
120349 2020-09
129666 2020-10
147158 2020-11
14211 2020-12
```

Du coup, on voit un **x30 en mars et avril 2020**.

Cela correspond au moment où Framasoft a réparti sur les chatons les créations de pad qui les submergeaient.

Le 21/04, j'ai fait demander par courriel à ce que pad.april.org soit retiré de leur liste de délestage et de privilégier pad.chapril.org. Réponse de validation le jour même.

On constate qu'après, on est en x10 puis x3. Il se peut qu'une partie des utilisateurs continuent de venir d'eux-mêmes.

Suggestion d'ajouter dans texte de nouveau pad une invitation à aller sur pad.chapril.org.

#7 - 05/12/2020 12:19 - François Poulain

Le confinement ;o)

Oui, bonne analyse.

Du coup je pense :

- 1) qu'on devrait virer les pad à l'abandon.
- 2) qu'on devrait virer (ou oublier l'historique) de pads qui ont beaucoup vécu.

Je pense que notre db et notre pad ont simplement des pb de montée en charge.

Enfin pour les session storage, il y a eu plein de bug chez etherpad ouverts sur le sujet. On peut imaginer qu'on traîne des zombies (plus de 1 millions de lignes en db, avec un requetage probablement pas optimal). D'où la proposition de sauver une liste de sessions et supprimer celles qui subsisteront dans qq mois.

#8 - 05/12/2020 12:32 - Christian P. Momon

François Poulain a écrit :

Autre chose. Si je prends les clés utilisées dans la table :

1 135 045 sessionstorage:

Et que je regarde les clés les plus utilisées, j'ai quelques suspects :
- des sessions qui traînent

On faisant une recherche dans le gestionnaire de tickets du projet officiel, je suis tombé sur :

- <https://github.com/ether/etherpad-lite/issues/3208>
 - sessionstorage entries don't get cleaned up
 - dupe of [#1659](#)

Et donc :

- <https://github.com/ether/etherpad-lite/issues/1659>
 - session is reset multiple times [Chrome Specific?]

En résumé :

- ça dit que plein d'entrées sessionstorage sont créées à tort et non nettoyée ;
- bug détecté en mars 2013 mais n'avait pas le temps de s'en occuper ("I'm not gonna have time to look at this for a bit"),
- correction le 03 avril 2020
 - <https://github.com/ether/etherpad-lite/commit/c2ea2b3a6d152da7f920cd9092ce4ec66d500a67>

```
webaccess: do not resave session
Before this change, the database was spammed with session values.
Modern express-session has this baked in.
See https://www.npmjs.com/package/express-session#resave for docs.
```

Le correctif est actif depuis la version 1.8.3 d'avril 2020 mais ne fait pas le nettoyage des lignes inutilement accumulées.

Conclusion : **proposition de faire un gros ménage des entrées sessionstorage qui représentent 1/8 des entrées totales** . Ça devrait alléger un peu la base de données :-)

#9 - 05/12/2020 12:37 - Christian P. Momon

François Poulain a écrit :

Enfin pour les session storage, il y a eu plein de bug chez etherpad ouverts sur le sujet.
On peut imaginer qu'on traîne des zombies (plus de 1 millions de lignes en db, avec un requetage probablement pas optimal).
D'où la proposition de sauver une liste de sessions et supprimer celles qui subsisteront dans qq mois.

Commençons par ça.

A priori, il n'y a pas de grosse incidence suite au vidage des entrées sessionstorage.
Donc choisissons une date pour l'opération et supprimons-les toutes.

#10 - 05/12/2020 22:25 - Christian P. Momon

Pour info, 94 Mo gagnables en purgeant les sessionstorage inutiles :

```
MariaDB [etherpad]> select sum(length(store.value)) AS SIZE from store where store.key like 'sessionstorage:%'
;
+-----+
| SIZE  |
+-----+
| 98751237 |
+-----+
```

#11 - 05/12/2020 23:03 - Christian P. Momon

- Fichier statpadaprilorgdb ajouté

#13 - 07/12/2020 19:15 - François Poulain

NB: Je ne sais pas si c'est mieux mais on peut brancher un redis sur etherpad.
<https://github.com/ether/etherpad-lite/wiki/How-to-use-Etherpad-Lite-with-Redis>

Etherpad utilise mysql pour une base clé/valeur. C'est vraiment un mauvais cas de figure pour sql. On peut facilement imaginer que l'usage d'une db clé/valeurs comme redis apporte du mieux en perfs.

#14 - 07/12/2020 19:16 - François Poulain

Sachant que à l'inverse, mysql est notoirement connu pour charger les IO disques ; là où redis est notoirement connu pour travailler d'abord en RAM.

#15 - 07/12/2020 19:49 - Christian P. Momon

François Poulain a écrit :

Etherpad utilise mysql pour une base clé/valeur. C'est vraiment un mauvais cas de figure pour sql.

+1

Sachant que à l'inverse, mysql est notoirement connu pour charger les IO disques

+1

là où redis est notoirement connu pour travailler d'abord en RAM.

-1 car j'avais cru comprendre que Redis chargeait **TOUT** en RAM. Extrait de <https://fr.wikipedia.org/wiki/Redis> :

Une des principales caractéristiques de Redis est de conserver l'intégralité des données en RAM. Cela permet d'obtenir d'excellentes performances en évitant les accès disques, particulièrement coûteux sur ce plan.

Quand la taille des données est trop importante pour tenir en mémoire, Redis peut également utiliser de la mémoire virtuelle.

Afin d'assurer la conservation des données en cas d'incident – la mémoire vive étant volatile – Redis offre la possibilité de « capturer » l'état de la base dans un fichier. Cette technique ne conservant pas les modifications effectuées entre deux captures, il est par ailleurs possible de les enregistrer afin de restaurer la base en cas d'incident.

Donc :

- Redis semble être plus un cache qu'une base de données :-/
- faudrait attribuer 6 Go de RAM à la vm ?
- faut accepter une persistance pas très garantie...

Je ne suis pas très rassuré. Et vous ?

#16 - 07/12/2020 19:55 - François Poulain

<https://redis.io/topics/persistence>

NB: je n'ai pas d'expérience en redis.

#17 - 30/12/2020 22:07 - Quentin Gibeaux

- Version cible changé de Décembre 2020 à Janvier 2021

#18 - 27/01/2021 21:31 - Quentin Gibeaux

- Version cible changé de Janvier 2021 à Février 2021

#19 - 29/01/2021 20:26 - François Poulain

J'ai sauvé la liste des sessions :

```
mariadb -u pad -prouth etherpad -e 'select store.key from store where store.key like "sessionstorage:%" > 20200129_sessionstorage.dat
```

#20 - 29/01/2021 20:26 - François Poulain

Prochaine action vers la fin du mois : prendre des salves de clé et deleter les sessions.

#21 - 24/02/2021 21:43 - Quentin Gibeaux

- Version cible changé de Février 2021 à Mars 2021

#22 - 06/03/2021 15:13 - François Poulain

Je fais ainsi :

```
cp 20200129_sessionstorage.dat 20200129_sessionstorage.sql
sed -i -e 's/^/DELETE FROM store WHERE store.key = "/" -e 's/$/";/' 20200129_sessionstorage.sql
mariadb -u pad -prouth etherpad < 20200129_sessionstorage.sql
```

C'est long. :)

#23 - 06/03/2021 18:34 - François Poulain

J'abandonne car un strace me dit au bout de plusieurs heures que j'en suis à 22k requetes sur 1 million et qu'elles ne passent pas.

```
recvfrom(3, "\7\0\0\1\0\1\0\2\0\0\0", 16384, MSG_DONTWAIT, NULL, NULL)
= 11
sendto(3, "\v\0\0\0\3DELETE FROM store WHERE store.key = \"sessionstorage:0Cq5jx8ulCswDjVgOVQM74a\"...", 90, MSG_DONTWAIT|MSG_NOSIGNAL, NULL, 0) = 90
recvfrom(3, 0x55aaab9404d0, 16384, MSG_DONTWAIT, NULL, NULL)
```

```
    = -1 EAGAIN (Resource temporarily unavailable)
poll([{fd=3, events=POLLIN}], 1, -1)
    = 1 ( [{fd=3, revents=POLLIN}])
```

Je vais essayer une autre stratégie.

#24 - 06/03/2021 18:58 - François Poulain

C'est passé pendant l'apéro :

```
split ../20200129_sessionstorage.dat
for file in * ; do mariadb etherpad -e "DELETE FROM store WHERE store.key in ( $(cat $file | sed -e 's/^/"/' -
e 's/$/"/' ) )"; done
```

#25 - 06/03/2021 19:02 - François Poulain

Bof, pas convaincu :

```
MariaDB [etherpad]> select count(value) from store where store.key like 'sessionstorage:%' ;
+-----+
| count(value) |
+-----+
|          1196501 |
+-----+
1 row in set (17.254 sec)
```

MariaDB [etherpad]>

#26 - 06/03/2021 19:16 - François Poulain

Curieux :

```
MariaDB [etherpad]> SELECT store.key FROM store WHERE store.key IN ( "sessionstorage:H3PYMvtn1IIMFHDRsfnNS01neEhSgwCJ" "sessionstorage:H3PfKEo0hZmMgAiR33qJyfk4pVkjkcOK" "sessionstorage:H3PiXecEu_KA8hCOPjnmyoo9Kw77Grer" )
;
Empty set (0.000 sec)
```

```
MariaDB [etherpad]> SELECT store.key FROM store WHERE store.key = "sessionstorage:H3PYMvtn1IIMFHDRsfnNS01neEhSgwCJ" ;
+-----+
| key |
+-----+
| sessionstorage:H3PYMvtn1IIMFHDRsfnNS01neEhSgwCJ |
+-----+
1 row in set (0.001 sec)
```

#27 - 06/03/2021 19:27 - François Poulain

Arf, il faut une virgule dans la syntaxe.

Donc la commande qui fonctionne :

```
for file in * ; do mariadb etherpad -e "DELETE FROM store WHERE store.key in ( $(cat $file | sed -e 's/^/"/' -
e 's/$/"/' | tr '\n' ' ' | sed -e 's/" "/,"/g') );"; done
```

#28 - 06/03/2021 19:37 - François Poulain

Victoire :

```
MariaDB [etherpad]> select count(value) from store where store.key like 'sessionstorage:%' ;
+-----+
| count(value) |
+-----+
|          36462 |
+-----+
1 row in set (0.120 sec)
```

#29 - 06/03/2021 19:46 - François Poulain

À l'issue de ça le dump de la db, gzippé, fait encore 571M.

#30 - 06/03/2021 19:55 - François Poulain

Bon : ce qu'on a gagné est rikiki :

```
(April) root@pad:~/20200129_sessionstorage# ls -lh /var/backups/mysql/
total 237M
-rw-r--r-- 1 root root 237M mars 6 05:43 etherpad.sql.bz2
-rw-r--r-- 1 root root 97K mars 6 05:43 mysql.sql.bz2
(April) root@pad:~/20200129_sessionstorage# /etc/rrsync.d/dump-mysql
(April) root@pad:~/20200129_sessionstorage# ls -lh /var/backups/mysql/
total 210M
-rw-r--r-- 1 root root 210M mars 6 19:54 etherpad.sql.bz2
-rw-r--r-- 1 root root 97K mars 6 19:54 mysql.sql.bz2
```

#31 - 06/03/2021 23:31 - Christian P. Momon

Belle différence quand même :

Clés sessionstorage	Lignes	Taille (octets)
Avant	1 135 045	98 751 237
Après	36 544	3 325 577

Bravo \o/

#32 - 31/03/2021 21:54 - Quentin Gibeaux

- Version cible changé de Mars 2021 à Avril 2021

#33 - 28/04/2021 21:31 - Quentin Gibeaux

migration vers postgresql ?

#34 - 28/04/2021 21:35 - Quentin Gibeaux

todo: migration vieux pad confinement

#35 - 28/04/2021 21:37 - Quentin Gibeaux

- Version cible changé de Avril 2021 à Mai 2021

#36 - 26/05/2021 21:39 - Quentin Gibeaux

- Version cible changé de Mai 2021 à Juin 2021

#37 - 30/06/2021 21:10 - Quentin Gibeaux

- Version cible changé de Juin 2021 à Été 2021

#38 - 01/09/2021 22:03 - Quentin Gibeaux

- Version cible changé de Été 2021 à Septembre 2021

#39 - 01/09/2021 22:30 - Quentin Gibeaux

- Version cible changé de Septembre 2021 à Été 2021

#40 - 01/09/2021 22:31 - Quentin Gibeaux

- Statut changé de En cours de traitement à Fermé

todo déplacé dans un autre ticket

Fichiers

statpadaprilorgdb	4,42 ko	05/12/2020	Christian P. Momon
-------------------	---------	------------	--------------------